Theoretical Computer Science 432 (2012) 58–63

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the complexity of the edge-disjoint min–min problem in planar digraphs☆

Longkun Guo[a], Hong Shen[b,c,∗]

[a] School of Computer Science and Mathematics, Fuzhou University, China
[b] School of Computer and Information Technology, Beijing Jiaotong University, China
[c] School of Computer Science, University of Adelaide, Australia

ARTICLE INFO

Article history:
Received 6 January 2011
Received in revised form 28 November 2011
Accepted 9 December 2011
Communicated by D.-Z. Du

Keywords:
Min–min problem
Planar digraph
NP-complete
Disjoint path
Inapproximability

ABSTRACT

The min–min problem of finding a disjoint path pair with the length of the shorter path minimized is known to be *NP*-complete (Xu et al., 2006) [1]. In this paper, we prove that in planar digraphs the edge-disjoint min–min problem remains *NP*-complete and admits no K-approximation for any $K > 1$ unless $P = NP$. As a by-product, we show that this problem remains *NP*-complete even when all edge costs are equal (i.e., strongly *NP*-complete). To our knowledge, this is the first *NP*-completeness proof for the edge-disjoint min–min problem in planar digraphs.

1. Introduction

The min–min problem of computing an edge-disjoint (vertex-disjoint) path pair with the length of the shorter path minimized has attracted considerable attention in the research community [1–3]. This problem is important in numerous practical applications of networks. In some critical real-time applications, to guarantee the robustness of the service provided, it is required to establish multiple vertex-disjoint (edge-disjoint) routes between a pair of nodes, so that, in case one of the paths fails, the other routes are still functioning. That is, an alternative path can be established immediately for any vertex or edge failure in the current path to ensure robust routing and a certain degree of fault tolerance in the network. In real networks, it is quite common that nodes or edges may fail. To protect a connection from a single edge (vertex) failure, there are two types of basic approach. One is called on-demand recovery, which invokes a routine for finding another shortest path when a vertex or edge failure on the current route is detected. This type of restoration works well for applications such as datagram communications in the Internet, but it does not suit mission-critical communications which cannot tolerate the long recovery latency required for finding an alternative path. The worst scenario is that there may not be such a path in the graph since there are not enough backup resources available at that moment. An alternative solution to this approach is known as preplanned failure restoration: the restoration process is done instantly because the backup path is predefined and its required resources have been reserved. Secondly, for safety or security reasons in network applications, two types of commodity (e.g., encrypted data and a decryption key) need to be transported separately along two disjoint paths. Because

☆ This work was partially supported by the "100 Talents" Project of Chinese Academy of China, National Science Foundation of China grant #61170232 and Fundamental Research Funds for the Central Universities. The corresponding author is Hong Shen.

∗ Corresponding author at: School of Computer and Information Technology, Beijing Jiaotong University, China. Tel.: +61 8 83035592.
E-mail address: hong@cs.adelaide.edu.au (H. Shen).

0304-3975/$ – see front matter © 2012 Published by Elsevier B.V.
doi:10.1016/j.tcs.2011.12.009

Table 1
Known complexities of problems of computing disjoint paths.

Problem		General graph	Planar graph		
[a] *Min-sum*		*Polynomially solvable [6]*			
[b] *Min-max*		*NP-complete [8,12]*			
[c] *2DP*	Undirected	*Polynomially solvable [9,10]*			
	Directed	*NP-complete [11]*	Edge-disjoint	*Open*	
			Vertex-disjoint	*Polynomially solvable [13]*	
[d] *Min–min*		*NP-complete [1,5,14]*	Edge-disjoint	Undirected	*Open*
				Directed	***NP-complete*** **[**This paper**]**
			Vertex-disjoint	*Polynomially solvable [14]*	

[a] *Min-sum*: Compute two disjoint paths from s to t with minimum total length.
[b] *Min-max*: Compute two disjoint paths from s to t with minimum length of the longer path.
[c] *2DP*: Compute two disjoint paths from s_1 to t_1 and s_2 to t_2.
[d] *Min–min*: Compute two disjoint paths with minimum length of the shorter path.

transporting one commodity (e.g., encrypted data) is much more expensive than the other one (e.g., decryption key), finding the shortest path with a disjoint counterpart is the major design objective. Third, solving the min–min problem can lead to near-optimal solutions to the well-known MSOD (min-sum in an ordered dual network) problem, which is the min-sum problem in a network in which each edge has two ordered "lengths" and the length of an edge used by the longer path is always a fraction of the length used by the shorter path [3,4]. Formally, this problem is defined as follows.

Definition 1. Given an undirected (directed) graph $G = (V, E)$, distinct vertices $s, t \in V$, and length function $f : E \to \mathbb{R}^+$, find a vertex-disjoint (edge-disjoint) path pair AP and BP with $f(AP)$ minimized, where AP is the *active* path, BP is the *backup* path, and $f(AP) \leq f(BP)$.

It is known that this problem is *NP*-complete [1,3,5]. The complexity of problems closely related to the min–min problem has been well studied. The min-sum problem of finding two paths with the total length minimized is polynomially solvable [6,7], whereas the min-max problem of finding two disjoint paths with the length of the longer path minimized is *NP*-complete [8]. The length-bounded disjoint path problem, to find two disjoint paths with the length of each path bounded by a given bound, is a variant of the min-max problem, and it is also known to be *NP*-complete [8]. Applying the algorithm for the min-sum problem in [6,7] will result in a 2-approximation solution for the min-max problem and the length-bounded disjoint path problem, achieving the best approximation ratio for these two problems in directed graphs [8].

For a given graph $G = (V, E)$ and four vertices $s_1, s_2, t_1, t_2 \in V$, the 2-disjoint path (2DP) problem is to compute two disjoint paths, one from s_1 to t_1 and another from s_2 to t_2. As a well-known result, the 2DP problem is polynomially solvable in undirected graphs [9,10], whereas it is *NP*-complete in directed graphs [11]. This problem is closely related to the min-max problem and the min–min problem, because, based on the *NP*-completeness of the directed 2DP problem, we can obtain the inapproximability of the min-max problem in directed graphs, i.e., the directed min-max problem admits no ρ-approximation solution for any $\rho < 2$ [8]. Following the same lines as the inapproximability proof in [8], clearly the directed min–min problem admits no K-approximation for any $K > 1$.

Because most practical applications arise in a network that can be mapped into a planar graph, the problems above have been considered in planar graphs. Holst et al. showed the *NP*-completeness of the length-bounded disjoint problem in planar graphs by giving a reduction from the *partition* problem [12]. The vertex-disjoint version of the well-known 2DP problem is polynomial solvable in planar digraphs [13], whereas the complexity of its edge-disjoint version remains a long-standing open problem. Fitting the complexity of the 2DP problem, the vertex-disjoint min–min problem is polynomial solvable in planar digraphs [14], while its edge-disjoint version is *NP*-complete in planar digraphs, as shown in this paper. Table 1 lists relevant disjoint-path problems and their known complexities.

This paper shows the *NP*-completeness of the edge-disjoint min–min problem in digraphs and extends the proof to show the *NP*-completeness and also the inapproximability of this problem in planar digraphs. We note that the *NP*-completeness of the edge-disjoint min–min problem in general digraphs was proved in [1,3]. However, our proof can be extended to show the complexity and the inapproximability of the edge-disjoint min–min problem in planar digraphs. To our knowledge, this is the first *NP*-complete proof for this problem in planar digraphs, and it may be the first step in answering the long-standing question whether the 2DP problem is *NP*-complete in planar digraphs.

2. *NP*-completeness proof for the edge-disjoint min–min problem in digraphs

In this section, we show the *NP*-completeness of the min–min problem in general digraphs. We first give some notation. Assume that P is a path from v_0 to v_h following the vertex order $v_0, v_1, \ldots, v_h, u, w \in V(P)$, where $u = v_i$, $w = v_j$, and $i < j$. We denote by $e(w, u)$ the edge from w to u, and call $e(w, u) \notin P$ a backward edge if $u \prec_P w$.

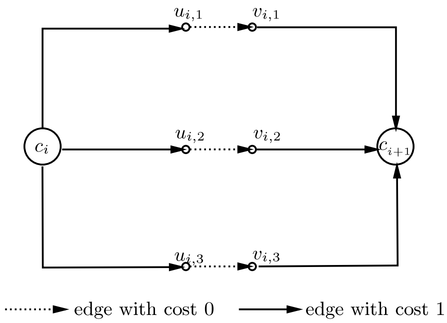

Fig. 1. Lobe for C_i.

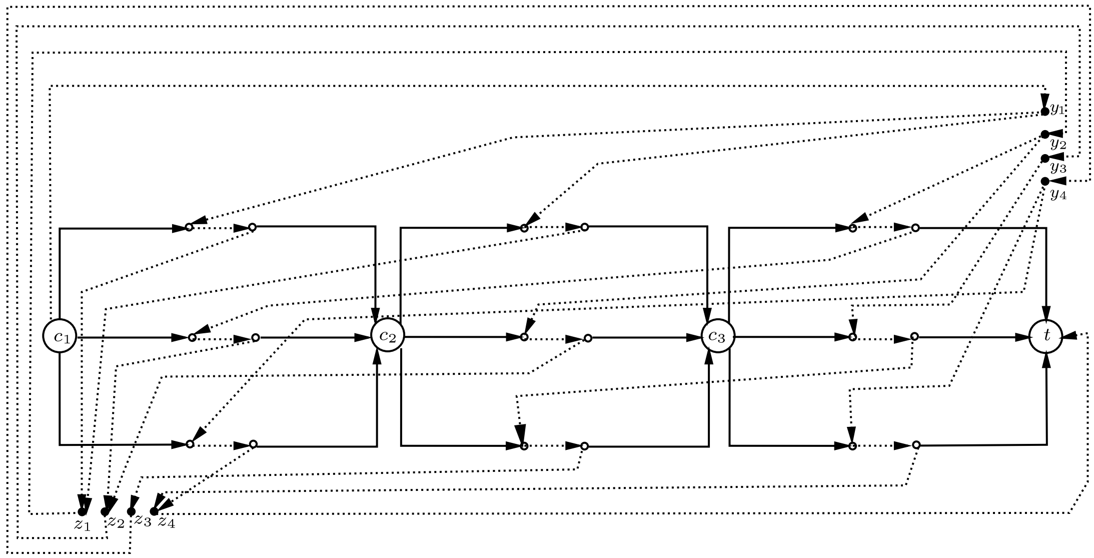

Fig. 2. Reduction of the *3SAT* instance $x_1 \vee x_2 \vee \bar{x}_4, \bar{x}_1 \vee \bar{x}_2 \vee x_3, x_2 \vee x_3 \vee x_4$ in digraphs.

By reducing the well-known *NP*-complete *3-Satisfiability (3SAT)* problem, we shall prove the following theorem.

Theorem 1. *The edge-disjoint min–min problem in digraphs is* NP-*complete.*

Proof. We prove the theorem by reducing *3SAT* to the decision form of our problem: Does a digraph with positive edge cost contain two edge-disjoint s–t paths *AP* and *BP* with $f(AP) = 0$?

An instance of *3SAT* is *AND* of m clauses $C_1 \wedge \ldots \wedge C_m$, where $C_i = a_{i,1} \vee a_{i,2} \vee a_{i,3}$, and $a_{i,j}$ is an occurrence of a variable in $X = \{x_1, \ldots, x_n\}$ or its negation. For a given instance of *3SAT*, we construct an auxiliary graph such that the instance of *3SAT* is satisfiable iff there exist two disjoint paths. The construction is as follows.

First, for a clause C_i we add a vertex c_i and construct a lobe G_i (as illustrated in Fig. 1), which is three paths from c_i to c_{i+1} with three cost-0 middle edges $e(u_{i,j}, v_{i,j})$ $(j = 1, 2, 3)$, respectively.

Second, for a variable x_j, we add two vertices y_j and z_j together with dotted edges (cost-0 edges) $s \rightarrow y_1, z_j \rightarrow y_{j+1}$ $(j = 1, \ldots, n-1)$, and $z_n \rightarrow t$. We connect those vertices representing variables via dotted edges as follows. Assume that literals $a_{j_1,l_1}, \ldots, a_{j_h,l_h}$ with $j_1 < j_2 < \ldots < j_h$ are occurrences of x_j incrementally, then add backward edges $y_j \rightarrow u_{j_h,l_h}, \ldots, v_{j_i,l_i} \rightarrow u_{j_{i-1},l_{i-1}}, \ldots, v_{j_1,l_1} \rightarrow z_j$ (the case for $\bar{x}_j$ is similar).

For example, graph G constructed for the instance $x_1 \vee x_2 \vee \bar{x}_4, \bar{x}_1 \vee \bar{x}_2 \vee x_3, x_2 \vee x_3 \vee x_4$ is as illustrated in Fig. 2. Apparently, the construction of G can be finished in polynomial time. So Lemma 1 below completes the proof of this theorem. □

Lemma 1. *There exists a true assignment satisfying the instance of* 3SAT *iff there exist two edge-disjoint min–min paths in G with* $f(AP) \leq 0$.

Proof. We show that the satisfiability of the instance results in the existence of a solution of the problem, and vice verse.

Suppose that there is a true assignment satisfying the instance of *3SAT*. The min–min paths pair *AP* and *BP* can be constructed as follows. For *AP*, initially $s \rightarrow y_1$ and $z_n \rightarrow t$ are added. Assume that P_j^1 and P_j^2 are two paths from y_j to z_j corresponding to x_j and $\bar{x}_j$, respectively. Then, for each $i \in \{1, \ldots, n\}$, add P_i^1 to *AP* if $\tau(x_i) = $ *false* and add P_i^2 to *AP* otherwise. Apparently, *AP* is a cost-0 path from s to t. Because each clause C_i contains at least one satisfied literal, every lobe must contain at least one path that shares no common edge with AP. Then, there are n paths of the n lobes that collectively compose a path BP from s to t disjoint with AP.

Conversely, assume that AP and BP are two min–min paths in G and $f(AP) = 0$. The true assignment for each x_j is as follows. For $a_{i,l}$ appearing on BP, set $\tau(x_j) = true$ if $a_{j,l}$ is an occurrence of x_j, and $\tau(x_j) = false$ otherwise. To show that this true assignment works correctly, we need to show first that for any x_j either P_j^1 or P_j^2 shares no common edge with BP, and second that for each lobe BP contains at least one of its three paths.

For the first, since $f(AP) \leq 0$ and every solid edge has cost 1, AP cannot contain any solid edge. Hence, AP must contain every edge that can separate s and t in G after removing the solid edges. Therefore, AP must contain $s \to y_1, \dots, z_j \to y_{j+1}, \dots, \to y_n,\ z_n \to t$. Because the dotted edges do not compose any loop, AP will go through y_j and z_j incrementally on j. Since there exist only two cost-0 paths P_j^1 and P_j^2 from y_j to z_j, AP must go through either P_j^1 or P_j^2. From the fact that AP and BP are edge disjoint, BP cannot go through any edge appearing on AP. So either P_j^1 or P_j^2 shares no common edge with BP.

For the second, BP must go through every c_i since c_i separates s and t in $G \setminus E(AP)$. Because BP is a path, BP cannot contain any loop. Since each backward edge on BP indicates a loop, BP contains no backward edges. Therefore, BP must go through one of the three paths in every lobe. These complete the proof. □

Our *NP*-complete proof for the edge-disjoint min–min problem in general digraphs is simpler than Xu et al.'s proof in [1], and can be extended to the problem in planar digraphs, which is the main contribution of our paper.

3. Extended proof to planar digraphs

In this section, we show that our *NP*-completeness proof can be extended to planar graphs. Note that the auxiliary graph G constructed in Section 2 may not be a planar graph, since G may contain some cross edges that cannot be embedded into a plane (i.e., $K_{3,3}$ or K_5). Observe that, for an edge e which cannot be embedded into the plane, adding a dummy vertex to G on the point at which e cross another edge will decrease the number of the edges that cross e. The main idea of our proof is to add a set of proper dummy vertices to G, such that the resulting graph H is a planar graph which contains a min–min path pair AP and BP with $f(AP) = 0$ iff the *3SAT* instance is satisfiable. To do this, we add at most two extra vertices to every path of a lobe for each P_g^l ($g = 1, \dots, n;\ l = 1, 2$). The construction of the auxiliary graph H for a given *3SAT* instance is as follows (for example, graph H constructed for instance $x_1 \vee x_2 \vee \bar{x}_4, \bar{x}_1 \vee \bar{x}_2 \vee x_3, x_2 \vee x_3 \vee x_4$ is as illustrated in Fig. 4).

Algorithm 1. Construction of the auxiliary graph for an instance of *3SAT*

Input:
$X = \{x_1, \dots, x_n\}$: variables of a *3SAT* instance;
$C = \{C_1, \dots, C_m\}$: $C_i = a_{i,1} \vee a_{i,2} \vee a_{i,3}$, where $a_{i,j}$ is an occurrence of x_l (w.l.o.g., assume that $\{C_1, \dots, C_m\}$ are sorted in alphabetic order);
Output:
H: The auxiliary graph corresponding to the given 3*SAT* instance.

1. Construct auxiliary graph G as in Section 2;
2. For each $a_{i,j} \in P_g^l$, add auxiliary vertices $b_{i,j,*}$ and $b'_{i,j,*}$ in the following order (as depicted in Fig. 3):

$$b_{i,j,2n}, \dots, b_{i,j,2g+l-2}, v'_{i,j}, u_{i,j}, v_{i,j}, b'_{i,j,2g+l-2}, \dots, b'_{i,j,2n}.$$

3. For $g = 1$ to n do
   For $l = 1$ to 2 do
   Assuming $P_g^l = y_g \to u_{j_g,l_g} \to \dots \to v_{j_i,l_i} \to u_{j_{i-1},l_{i-1}} \to \dots \to v_{j_1,l_1} \to z_g$, we replace $y_g \to u_{j_g,l_g}$, $v_{j_i,l_i} \to u_{j_{i-1},l_{i-1}}$ ($i = g, \dots, 2$) and $v_{j_1,l_1} \to z_g$ by three dotted subpaths as below, respectively (note that $b_{j,0,*} = b_{j+1,3,*}$):

$$y_g \to b'_{m,1,2g+l-2} \to \dots \to b'_{m,3,2g+l-2} \to b_{m,3,2g+l-2} \to \dots \to b_{j_g,l_g-1,2g+l-2} \to u_{j_g,l_g},$$
$$v_{j_i,l_i} \to v'_{j_i,l_i} \to b'_{j_i,l_i,2g+l-2} \to \dots \to b_{j_{i-1},l_{i-1}-1,2g+l-2} \to u_{j_{i-1},l_{i-1}},$$
$$v_{j_1,l_1} \to v'_{j_1,l_1} \to b'_{j_1,l_1,2g+l-2} \to \dots \to b_{1,1,2g+l-2} \to z_g.$$

Lemma 2. *The auxiliary graph H resulting from Algorithm 1 contains a pair of min–min paths AP and BP with $f(AP) = 0$ iff the instance of* 3*SAT is satisfiable.*

Proof. Similar to Theorem 1, so omitted. □

It remains to show the following lemma.

Lemma 3. *Graph H resulting from Algorithm 1 is planar.*

Proof. We prove the planarity of H by showing first that the dotted edges of H compose a planar graph, and second that the solid edges of H can be embedded without breaking the planarity of the dotted edges. From Algorithm 1, P_g^1 and P_g^2 are two paths from y_g to z_g corresponding to x_g and $\bar{x}_g$, respectively.

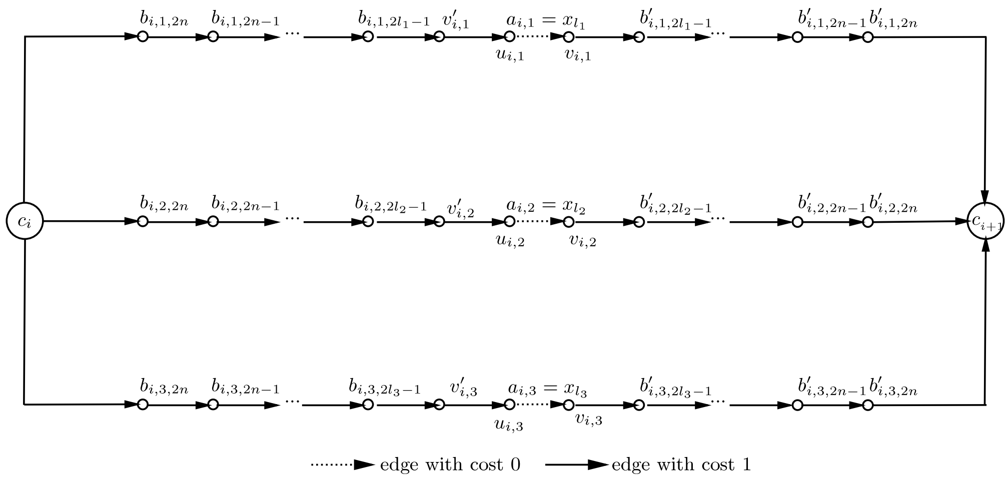

Fig. 3. Lobe for x_i in planar digraphs.

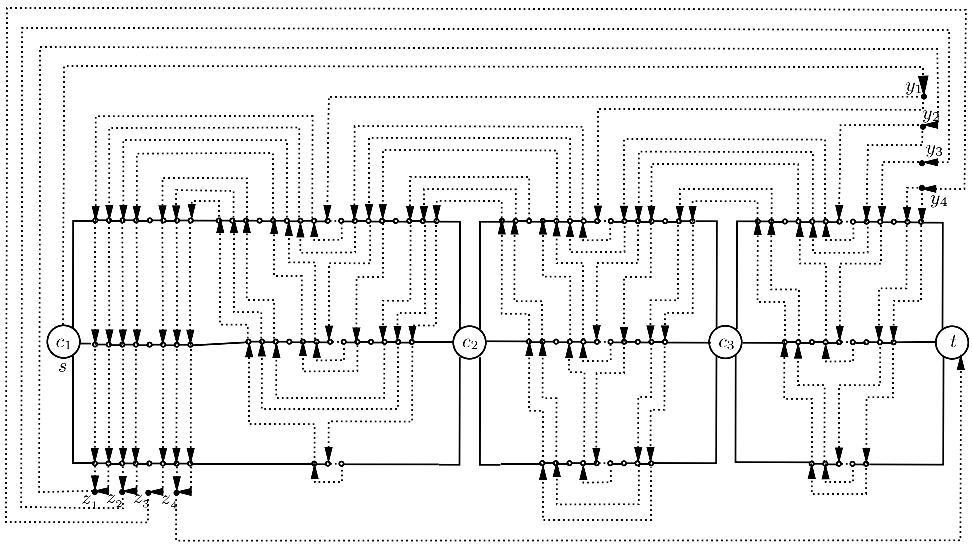

Fig. 4. Reduction of *3SAT* instance $(x_1 \vee x_2 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4)$ in a planar digraph.

For the first, the cost-0 edges (dotted edges) exactly compose the following paths: $Z = \{P(s, y_1),\ P_1^1(y_1, z_1),\ P_1^2(y_1, z_1),\ P(z_1, y_2), \ldots, P(z_n, t)\}$. We shall show these paths can be embedded into a plane. From the construction of H, these paths are pairwise interior vertex disjoint. Therefore, a pair of paths between an identical vertex pair is analogous to a pair of parallel edges. Let Z' be Z with every parallel path (edge) pair replaced by an edge connecting the same vertex pair. From [15], Z is a planar graph iff Z' is a planar graph. Since every vertex of $Z' \setminus \{s, t\}$ has degree 2 and $s,\ t$ have degree 1, Z' is a path between s and t. So the paths in Z compose a planar graph (as depicted in Fig. 5).

For the second, let $\mathbb{Y}$ be $Z \cup \{e(z_i, z_{i+1}), e(y_i, y_{i+1}) | i = 1, \ldots, n-1\}$ embedded in a plane in a way similar to Fig. 5. Then $\mathbb{Y}$ decomposes the plane to $2n-1$ bounded faces and one unbounded face. We need only to show that the solid edges can be embedded in the faces of $\mathbb{Y}$. Let P_{2i+l-2} be $P_i^l(y_i, z_i)$, and let Y_j be the face containing P_j and P_{j+1} on its boundary. Following Algorithm 1, there are two types of solid edge: those joining s, $\{c_2, \ldots, t\}$ and vertices on P_1 and P_{2n}, respectively, and those joining vertices of $P_i \cup P_{i+1}$. Clearly, edges of the first type can be embedded into the unbounded face of $\mathbb{Y}$. Hence only edges of the second type, denoted by $E_{i,i+1}$, remain, and we shall show they can be embedded into the face Y_i. From Algorithm 1, the solid edges of $E_{i,i+1}$ satisfy the following two conditions.

- For any two edges $e(u_1, v_1)$ and $e(u_2, v_2)$ with $u_1, u_2 \in P$ and $v_1, v_2 \in Q$, $u_1 \prec_P u_2$ (u_1 precedes u_2 on path P) holds iff $v_1 \prec_Q v_2$.
- For any edge $e(u_1, v_1)$ with $u_1, v_1 \in P$, H contains no $e(u_2, v_2)$ with $u_1 \prec_P x \prec_P v_1$ for $x \in \{u_2, v_2\}$.

Following the method of [15], clearly such solid edges of $E_{i,i+1}$ can be embedded into the face of Y_i. This completes the proof. $\square$

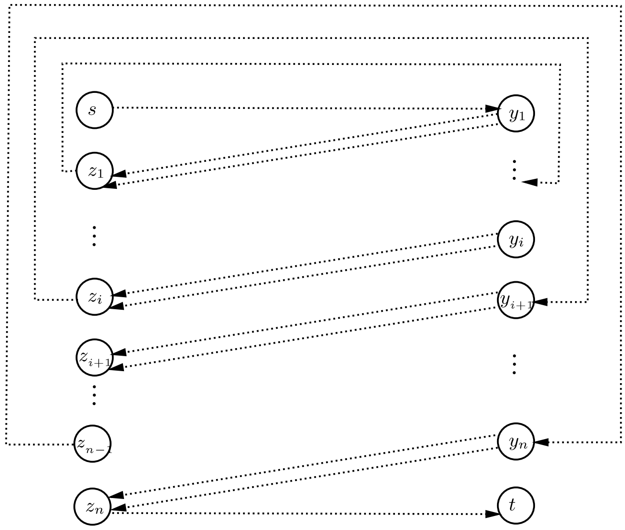

Fig. 5. The dotted edges embedded into a plane.

From the proof of Lemma 1, the edge-disjoint min–min problem admits no K-approximation for any $K > 1$. That is because any K-approximation for the problem can determine whether there exists a pair of edge-disjoint paths $\{AP, BP\}$ with $f(AP) = 0$ in G, yielding a polynomial solution for the *3SAT* problem, which is impossible unless *P=NP*. The case is similar in planar digraphs. Then, combining Lemmas 2 and 3, we immediately have the following theorem.

Theorem 2. *The edge-disjoint min–min Problem is* NP*-complete and admits no K-approximation for any $K > 1$ in planar digraphs.*

The correctness of the following corollary can be obtained from the proof of Lemma 3 by replacing each solid edge by $n + 1$ dotted edges and setting each dotted edge in H with cost 1.

Corollary 1. *The edge-disjoint min–min problem is strongly* NP*-complete in planar digraphs.*

4. Conclusion

In this paper, we have proved the *NP*-completeness of the edge-disjoint min–min problem in digraphs, and extended the proof to show the *NP*-completeness of this problem in planar digraphs. We achieved the latter by constructing a planar digraph H for any given instance of *3SAT*, such that the *3SAT* instance is satisfiable iff there is an edge-disjoint min–min path pair AP and BP in H with $f(AP) = 0$.

We are currently investigating how to extend our techniques to solving the edge-disjoint min–min problem in undirected planar graphs, whose complexity still remains open.

References

[1] D. Xu, Y. Chen, Y. Xiong, C. Qiao, X. He, On the complexity of and algorithms for finding the shortest path with a disjoint counterpart, IEEE/ACM Transactions on Networking 14 (1) (2006) 147–158.
[2] S. Zheng, B. Yang, M. Yang, J. Wang, Finding Minimum-Cost Paths with Minimum Sharability, in: IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications, 2007, pp. 1532–1540.
[3] C. Li, S. Thomas McCormick, D. Simchi-Levi, Finding disjoint paths with different path-costs: complexity and algorithms, Networks 22 (7) (1992).
[4] E. Bouillet, J. Labourdette, R. Ramamurthy, S. Chaudhuri, Lightpath re-optimization in mesh optical networks, IEEE/ACM Transactions on Networking (TON) 13 (2) (2005) 437–447.
[5] R. Bhatia, M. Kodialam, T. Lakshman, Finding disjoint paths with related path costs, Journal of Combinatorial Optimization vol. 12 (1) (2006) 83–96.
[6] J. Suurballe, Disjoint paths in a network, Networks 4 (2) (1974).
[7] J. Suurballe, R. Tarjan, A quick method for finding shortest pairs of disjoint paths, Networks 14 (2) (1984).
[8] C. Li, T. McCormick, D. Simchi-Levi, The complexity of finding two disjoint paths with min-max objective function, Discrete Applied Mathematics 26 (1) (1989) 105–115.
[9] P. Seymour, Disjoint paths in graphs, Discrete mathematics 306 (10-11) (2006) 979–991.
[10] Y. Shiloach, A polynomial solution to the undirected two paths problem, Journal of the ACM (JACM) 27 (3) (1980) 445–456.
[11] S. Fortune, J. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem* 1, Theoretical Computer Science 10 (2) (1980) 111–121.
[12] H. van der Holst, J. de Pina, Length-bounded disjoint paths in planar graphs, Discrete Applied Mathematics 120 (1-3) (2002) 251–261.
[13] A. Schrijver, Finding k disjoint paths in a directed planar graph, SIAM J. Computing 23 (4) (1994) 780–788.
[14] G. Longkun, S. Hong, On Finding Min–Min disjoint paths, accepted by Algorithmica.
[15] J. Hopcroft, R. Tarjan, Efficient planarity testing, Journal of the ACM (JACM) 21 (4) (1974) 549–568.